Tutorial: Building BlackBerry application to receive BIS PUSH notifications

Includes code snippets for onboarding and push listener

Author

Buddha Puneeth N.

Contents

What is PUSH Notification?

Push, or server push, describes a style of Internet-based communication where the request for a given transaction is initiated by the publisher or central server. It is contrasted with pull, where the request for the transmission of information is initiated by the receiver or client.

What is BIS?

BIS (BlackBerry Internet Service) is an email and synchronization service provided by RIM for BlackBerry users. BIS was created for BlackBerry users without an enterprise email account on a BlackBerry Enterprise Server (BES). BIS allows you to retrieve email from multiple POP3, IMAP, and Outlook Web App (OWA) on your BlackBerry, and synchronize your contacts, calendar, and deleted items from some email providers.

How BIS is different from BES?

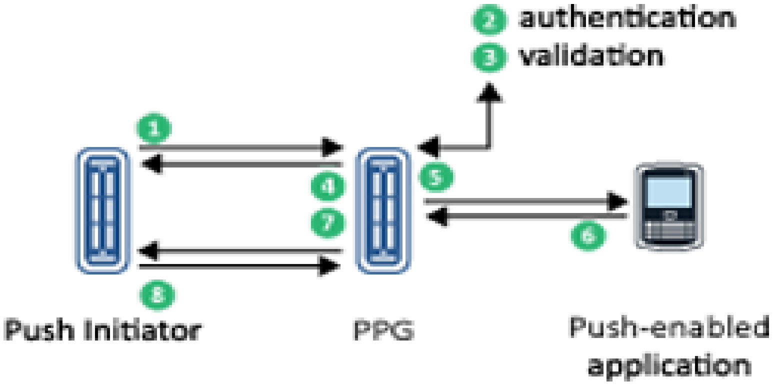
Both BES and BIS allow your BlackBerry to get email, as well as retrieve webpages and use third-party applications. All traffic from your phone goes to the BES / BIS server, and then the BES / BIS server communicates with the world on your behalf.

The difference, ultimately, lies in where the server resides and the level of control and security you get.

In the case of BIS, your carrier operates the server. Everything from BIS to your handheld is encrypted, but that’s about the extent of the security features. The carrier can decide what applications run on your phone, and how applications communicate with the Internet.

For BES, your company operates the server, and usually has it sitting somewhere within the corporate network. The IT department controls all aspect of the BES server, and it’s likely sitting in a nice and secure location.

Overview on BIS PUSH



BIS request response flow diagram

When the PPG is the BlackBerry Internet Service, push messages are sent following this process:

1. The Push Initiator sends a push message to the BlackBerry Internet Service in the form of an HTTPS POST.

   The push message is a MIME multipart message, which contains the following items:

- A WAP PAP 2.1 XML control entity, which describes the delivery parameters and specifies one or more BlackBerry devices to which the content will be delivered
- The content to deliver to the specified BlackBerry devices

2. The BlackBerry Internet Service receives the push message and attempts to authenticate the Push Initiator.
3. If the Push Initiator is successfully authenticated, the BlackBerry Internet Service validates the push message.

- If the Push Initiator is within the daily quota of push messages, and the request contains all the required information, the BlackBerry Internet Service accepts the push message.
- If the Push Initiator has exceeded the push limits, the BlackBerry Internet Service rejects the message.

4. The BlackBerry Internet Service returns a push response to acknowledge receipt of the push message, and indicates whether the message is accepted for processing or rejected. If the push message is rejected, the BlackBerry Internet Service returns an error code to the content provider that provides the reason for the rejection.
5. The BlackBerry Internet Service sends the push message to the specified BlackBerry devices.

6. Each BlackBerry device notifies the BlackBerry Internet Service when the push message is received. A push message is considered successful if the message is delivered before the specified expiry time and it meets the criteria specified by the <quality-of-service>element in the push message.
7. If the Push Initiator requests notification, the BlackBerry Internet Service sends a result notification (acknowledgement) message to the push server.
8. The Push Initiator responds to the BlackBerry Internet Service, acknowledging the receipt of the result notification.

To push data to the BlackBerry devices, two applications are required: a server-side application (push initiator) which submits the push request and a client side application which listens to the push messages.

Registering for BIS

To use the push service using BIS, we must register with the Research In Motion. Based on our type of registration we can avail the options from BlackBerry. One such option is level of service.

Levels of service

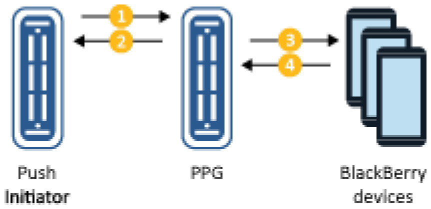
When you use the Push Service with the BlackBerry Internet Service, you can choose from two levels of service: Push Essentials and Push Plus.

When you use the Push Service with the BlackBerry Enterprise Server or the BlackBerry Device Service, the Push Plus level of service is included.

Push Essentials

When you use the Push Essentials level of service, you can send content to a device, but you can't query the status of the content or receive acknowledgments after you send the content. However, you can set the expiry time for content delivery for up to 30 days.

Here's how content is sent to a push-enabled application on a BlackBerry device with the Push Essentials level of service.



Flow in Push Essentials

1. The content provider sends content in a push request.

2. The PPG authenticates and acknowledges the push request.

3. The PPG pushes the content to the specified device or devices.

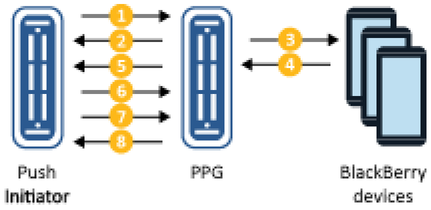
4. Each BlackBerry device acknowledges the push request to the PPG.

Push Plus

The Push Plus level of service offers the following additional functionality so that you can check the status and receive notifications about delivered content:

- Results notification lets you know when messages are delivered, if they couldn't be delivered, or if they expired waiting to be delivered.
- The PPG keeps track of the current status of a push message, and you can use a status query to check the status. The status of a push message can be pending, delivered, undeliverable, or expired.
- Push cancellation can withdraw messages if they haven't been delivered already.
- You can set the expiry time for content delivery for up to 8 hours.

Here's how content is sent to a push-enabled application on a BlackBerry device with the Push Plus level of service.



Flow in push plus

1. The content provider sends content in a push request, and can request to receive a result notification.

2. The PPG authenticates and acknowledges the push request.

3. The PPG pushes the content to the specified device or devices.

4. Each BlackBerry device acknowledges the push request to the PPG.

5. If the content provider requested to receive a result notification when it sent content in step 1, the PPG sends a notification to the content provider indicating whether the push message was delivered, was undeliverable, or expired waiting to be delivered.

6. The content provider acknowledges the receipt of the result notification.

7. Alternatively, if the content provider didn't request to receive a result notification, the content provider can request the status of the content from the PPG. The status request can be for any or all of the devices that received the content.

8. If the content provider requests the status of the content, the PPG returns a single response that contains the current status for the specified devices.

Once you decide on level of service, you can proceed with registration,

For trail usage you can register at https://www.blackberry.com/profile/?eventId=8121

For productive usage you can register at https://www.blackberry.com/profile/?eventId=8207

Once you register for PUSH services, you will receive a mail with following details: server URL, listener port, Application ID, Password and few more details.

Server side support

BlackBerry is providing Push Service SDK (Current version is 1.2), which provides Java APIs that Push initiator uses to interact with PPG (in our case it is BIS) and sends push messages to Push-enabled applications that run on BlackBerry devices. The Push Service SDK provides low-level and high-level APIs to handle the functionality in a push solution. Depending on your development environment and the design of the push solution, you might choose to use the low-level APIs to create PAP messages, and develop the rest of the solution on your own. Alternatively, you might want to use the high-level APIs to handle some of the more complex operations of your solution.

BIS PUSH using SMP

Overview of SMP

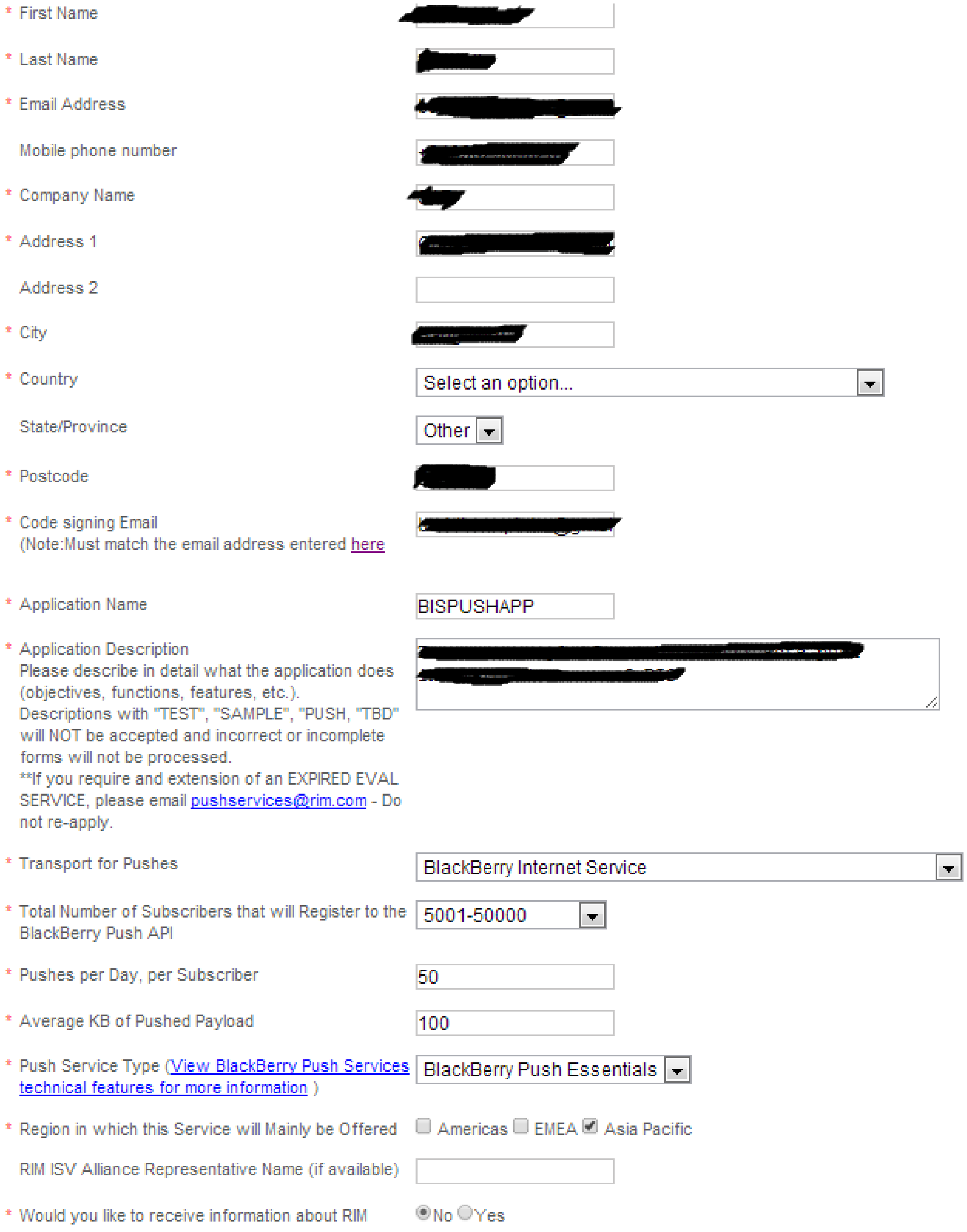
SMP3 unifies Sybase Unwired Platform, Sybase Mobiliser, Syclo Agentry and SAP's mobile technologies into one mobile platform for Business to Consumer (B2C) and Business to Employee (B2E) applications - embracing open standards such as HTML5, Cordova, OSGi Spring, OData and HTTP REST.

SMP will work as server side application in making PUSH requests. We need to do few configurations in order for BES PUSH to work from SMP side.

Before checking the PUSH configurations from SMP side, we need to register for BB PUSH services with BlackBerry.

Registering for BIS Service with BlackBerry:

1. Open Blackberry PUSH service page http://developer.blackberry.com/services/push/
2. Register to either evaluation service (6 months trail) or productive service, here we are giving level of service to 'Blackberry Push essentials'
3. Sample evaluation service looks like :



* First Name	[redacted]
* Last Name	[redacted]
* Email Address	[redacted]
Mobile phone number	[redacted]
* Company Name	[redacted]
* Address 1	[redacted]
Address 2	
* City	[redacted]
* Country	Select an option...
State/Province	Other
* Postcode	[redacted]
* Code signing Email (Note:Must match the email address entered here	[redacted]
* Application Name	BISPUSHAPP
* Application Description Please describe in detail what the application does (objectives, functions, features, etc.). Descriptions with "TEST", "SAMPLE", "PUSH, "TBD" will NOT be accepted and incorrect or incomplete forms will not be processed. **If you require and extension of an EXPIRED EVAL SERVICE, please email pushservices@rim.com - Do not re-apply.	[redacted]
* Transport for Pushes	BlackBerry Internet Service
* Total Number of Subscribers that will Register to the BlackBerry Push API	5001-50000
* Pushes per Day, per Subscriber	50
* Average KB of Pushed Payload	100
* Push Service Type (View BlackBerry Push Services technical features for more information)	BlackBerry Push Essentials
* Region in which this Service will Mainly be Offered	☐ Americas ☐ EMEA ☑ Asia Pacific
RIM ISV Alliance Representative Name (if available)	
* Would you like to receive information about RIM	● No ○ Yes

4. Once you register, you will get your service details after 5 working days, else you need to follow up with BB through mail: BlackBerryPushServices@blackberry.com
5. Your service will have following details:

BlackBerry Push Service Eval

Credentials
Registration Details
Company Name:
App Name:
Registered User Name:
Registered Email Address:

Account Administration Portal

Portal URL:
Portal Username:
Portal Password:

Server Configuration Details

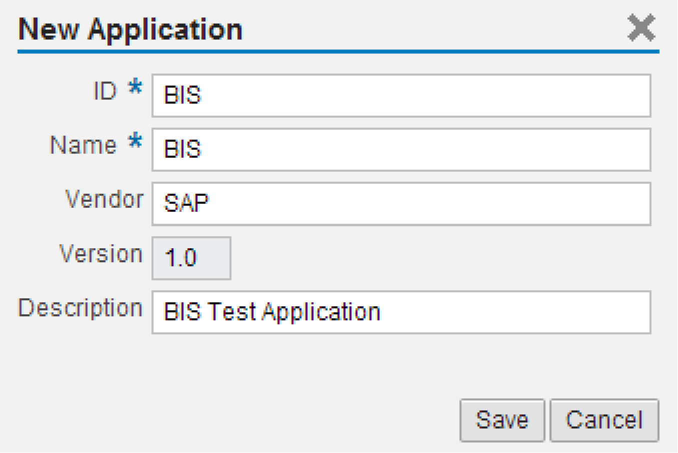
Application ID:
Push Password:
Content Provider ID (CPID):
Account Expiration Date:
Push URL:

Client Configuration Details

Application ID:
Push Registration URL:
Push Port*:

Configuring SMP for BIS PUSH

1. Open Admin portal and Under ‘Application’ tab, click on ‘New’ and provide necessary details



New Application

ID * BIS

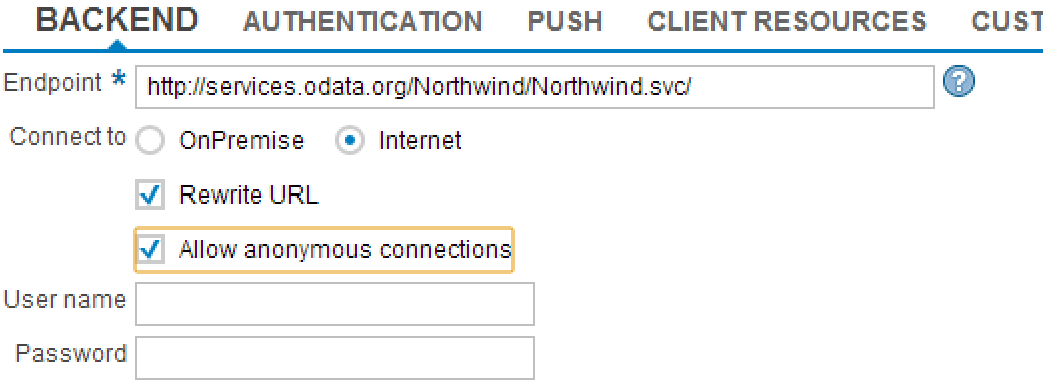
Name * BIS

Vendor SAP

Version 1.0

Description BIS Test Application

2. Provide Backend URL (Ebay OData Service) and Authentication details



Endpoint * http://services.odata.org/Northwind/Northwind.svc/

Connect to ○ OnPremise ● Internet

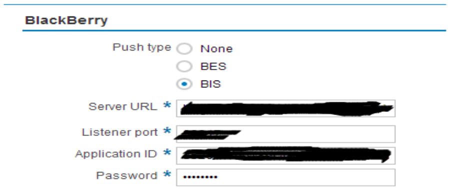
☑ Rewrite URL

☑ Allow anonymous connections

User name

Password

Configure with BIS push details under PUSH tab referring to the mail you received from BlackBerry regarding PUSH registration for App ID, Port and Password. Server URL remains common if you are using evolution service i.e., https://pushapi.eval.blackberry.com/mss/PD_pushRequest



BlackBerry

Push type ○ None ○ BES ● BIS

Server URL * [redacted]

Listener port * [redacted]

Application ID * [redacted]

Password * ••••••••

3. Click on SAVE
4. User should have PUSH role in order to perform PUSH operations, Admin has rights to provide this role for any SDN user

Client-side listener application

1. You can download the sample BIS listener application from http://developer.blackberry.com/bbos/java/documentation/download_build_sample_app.html
2. Source code for this is also available here, so that you can build your own application over this
3. In this tutorial, I'm using sample application as such for registering to BIS service and also for listening to incoming BIS PUSH messages. I'm enhancing the sample application with the onboarding code

Client-side Implementation

Implementation steps are as follows:

Registering to BIS service

1. Deploy the corresponding sample application to your device
2. Press Menu on your device, and click settings. The push configuration screen appears. Fill the details as shown below:
   a. Port : Pick form the BIS confirmation mail
   b. App ID : Pick form the BIS confirmation mail
   c. BPS Server URL : http://pushapi.eval.blackberry.com (in case of evaluation service)
   d. Push initiator application URL: https://pushapi.eval.blackberry.com/mss/PD_pushRequest (in case of evaluation service)
   e. Enable 'App Acknowledgement and Use public (BIS) network and SAVE
3. Press Menu on your device, and click register. It will ask you for username and password, in case of evaluation service no need to pass any credentials, and click on OK. If all the details provided are valid it will popup success message

Registering to SMP

You need to register to SMP , in onboarding body we need to pass the necessary details for BIS push like device type and device pin. The onboarding body looks like follows:

```
<?xml version='1.0' encoding='utf-8'?>
<entry xmlns="http://www.w3.org/2005/Atom"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
<title type="text"/>
<updated>2012-06-15T02:23:29Z</updated>
<category
term="applications.Connection"
scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"/>
<content type="application/xml">
<m:properties>
 <d:DeviceType>BlackBerry</d:DeviceType>
 <d:BlackberryDevicePin>XXXXXXXX</d:BlackberryDevicePin>
</m:properties>
</content>
</entry>
```

The sample onboarding request looks like:

```
HttpConnection httpCon; // httprequest declaration
                    OutputStream os;
                    rc = 0;
                    resp = "";
                    try {
                           // making request
                          httpCon = (HttpConnection)Connector.open("https<Cloud Base
URL>/odata/applications/latest/<App Name>/Connections", Connector.READ_WRITE);
                          // here BES is the name of SMP application
                           byte[] postDataBytes = body.getBytes(); //converting
sting into bytes
                           httpCon.setRequestMethod(HttpConnection.POST);
//Onboarding is always a post operation
                           httpCon.setRequestProperty("Authorization", "Basic
XXXXXXXXXX"); //Base64 encoded string of UN:PWD
                         httpCon.setRequestProperty("Content-
type","application/atom+xml");
                         os = httpCon.openOutputStream();
                         os.write(postDataBytes);  //making POST request with body as
argument
```

Initiating PUSH

Usually Backend will trigger the PUSH request when there is any change from its side. In this tutorial, we are simulating push request from the REST client.

Considering that user has PUSH role for this application.

> Sample PUSH URL: http(s) ://< Server Name> :< Run time Port>/Push/<APPCID>
> Headers:
> Authorization: Base-64 value of SDN credentials in format Username: Password
> x-sup-data or x-sup-rim-data or x-sap-poke-data: Data or details which we need to push

Receiving PUSH

In this tutorial, the sample PUSH application will take care of incoming PUSH messages to the device. You can see all the incoming notifications by clicking on 'View messages' in the start screen of the application.

References

1. BIS overview:
   http://developer.blackberry.com/services/push/
2. Developing sample PUSH application
   http://developer.blackberry.com/bbos/java/documentation/download_build_sample_app.html
3. SMP Cloud help:
   https://help.hana.ondemand.com/mobile/frameset.htm?SMP_welcome.html

www.sap.com